

Requested Patent: JP10055339A

Title: ON-LINE BUSINESS PROCESSING SYSTEM ;

Abstracted Patent: JP10055339 ;

Publication Date: 1998-02-24 ;

Inventor(s):

YOSHIKAWA AKIHIRO; KUDO YUTAKA; HARUNA TAKAAKI;
KOIZUMI SHINOBU ;

Applicant(s): HITACHI LTD ;

Application Number: JP19960213453 19960813 ;

Priority Number(s): ;

IPC Classification: G06F15/00 ; G06F9/06 ;

Equivalents:

ABSTRACT:

PROBLEM TO BE SOLVED: To improve operability and also to attain easy-to-understand display with a rich power of expression by generating a GUI(graphical user interface) processing program to display a GUI via a 2nd computer to perform an interactive processing with a user and also to perform the communication processing with an on-line business program via a communication means. SOLUTION: An on-line business program generation means 8 of a host computer 1 performs the processing to display a defined screen on a host terminal equipment based on the screen definition information 6. Then the means 8 compiles a business program source 7 to perform the on-line business processing and also generates an on-line business program 9 by combining those processings. A processing message template 13 describes the processing procedure of a GUI processing program 40 at its every part. A GUI processing program generation means 30 combines the template 13 based on the information acquired from the information 6 to generate the program 40.

757 5800 0002 615

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-55339

(43) 公開日 平成10年(1998) 2月24日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/00	3 1 0		G 0 6 F 15/00	3 1 0 S
9/06	5 3 0		9/06	5 3 0 N

審査請求 未請求 請求項の数 2 O L (全 13 頁)

(21) 出願番号 特願平8-213453

(22) 出願日 平成8年(1996) 8月13日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 吉川 彰弘

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 工藤 裕

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 春名 高明

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 弁理士 磯村 雅俊

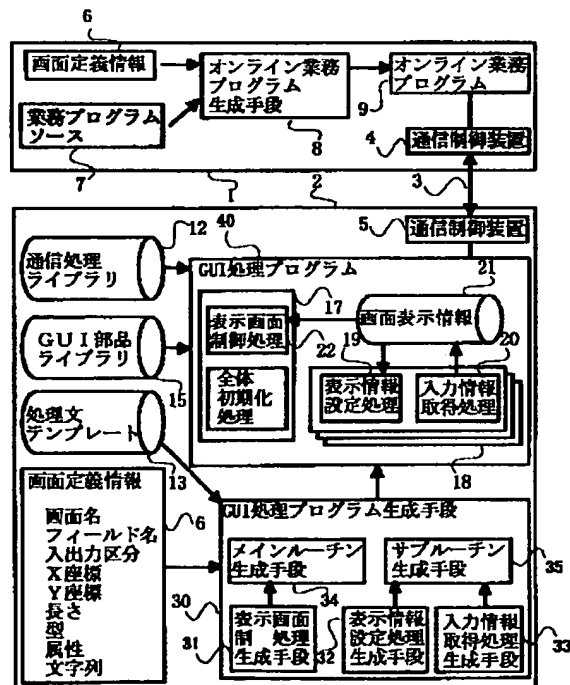
最終頁に続く

(54) 【発明の名称】 オンライン型業務処理システム

(57) 【要約】

【課題】 操作性の向上と豊かな表現力を用いた分かりやすい表示を行うためにGUIを用いてオンライン型業務処理を行うことを可能とするオンライン型業務処理システムを安価に提供すること。

【解決手段】 オンライン業務プログラムがホスト端末装置と通信するデータの形式が、ホスト端末装置の仕様によって決められているため、多くのオンライン業務プログラムに共通の形式であることに着目し、この形式を合わせるという手段により、既存のオンライン業務プログラムを修正せずにそのまま利用して、これと通信して動作するホストアクセスプログラムをGUI処理プログラムとして作成することを可能としたもの。



【特許請求の範囲】

【請求項1】 ホスト計算機となる第1の計算機と、通信手段と、キーボードおよびディスプレイを備えかつグラフィックス表示機能を有する第2の計算機とを有するオンライン型業務処理システムであって、

前記第1の計算機は、業務プログラムのソースコードと、オンライン型業務処理画面における画面中のデータフィールドのフィールド名称、表示座標、少なくとも入力か出力かを区別する入出力区分の対応表を定義する第1の画面定義情報とを有し、前記業務プログラムのソースコードと前記第1の画面定義情報とから、該第1の画面定義情報に従って前記通信手段に接続された他の装置に画面表示を行い、前記業務プログラムのソースコードに記述された処理手順に従ってオンライン業務処理を実行するオンライン業務プログラムを生成する手段を有し、前記第2の計算機は、前記第1の画面定義情報をコピーして得られる第2の画面定義情報と、該第2の画面定義情報から、GUIを表示して利用者との対話処理を行い、かつ、前記通信手段を介して前記オンライン業務プログラムとの通信処理を行うGUI処理プログラムを生成する手段を有することを特徴とするオンライン型業務処理システム。

【請求項2】 前記GUI処理プログラムを生成する手段は、前記第2の計算機上に、前記第2の画面定義情報の前記入出力区分が出力であるものについて、文字データの表示のための文字表示GUIを用い、前記表示座標で示される位置に表示する処理を生成する手段、前記第2の画面定義情報の前記入出力区分が入力であるものについて、文字データの入力のための文字入力GUIを用い、前記表示座標で示される位置から利用者の入力指示を受付る処理を生成する手段、利用者がデータ入力の終了を指示するためのボタンGUI処理を生成する手段、前記通信手段を介して前記オンライン業務プログラムからデータを受信して該文字表示GUIに表示する処理を生成する手段、および、前記ボタンGUIによってデータ入力の終了が指示されたときに、前記文字入力GUIに入力された文字データを、前記通信手段を介して前記オンライン業務プログラムに送信する処理を生成する手段を有することを特徴とする請求項1記載のオンライン型業務処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、通信手段を介してホスト計算機（汎用計算機、メインフレーム計算機とも呼ばれる）にアクセスして、業務処理を行うオンライン型業務処理システムに関し、特に、文字や罫線を主体とした、従来の使いにくい画面インタフェースに換えて、使いやすく美しいグラフィカル・ユーザインタフェース（以下、「GUI」という）を用いて業務処理を行うこ

とを可能とするオンライン型業務処理システムに関するものである。

【0002】

【従来の技術】 従来は、例えば、米国 Wall Data Inc.社のホスト・コネクティビティ・プロダクト「RUMBA」という製品の広告によれば、MicrosoftのWindowsアプリケーション開発ツールであるVisual Basicで利用可能なプログラム部品が提供され、これを用いて、ホスト計算機にアクセスするプログラムをVisual Basic言語で作成することができる。また、蝶理情報システム（株）のVisual Basic対応アプリケーション開発支援キット「EVB-SDK」という製品の広告によれば、ホスト計算機との通信接続やキー入力制御、キーシミュレーションなどを行う10種類のVisual Basic用プログラム部品が提供されている。

【0003】 また、これとは別に、（株）日立製作所の端末メッセージマッピング支援システム「XMAP 2E2」によれば、情報を画面表示するための領域の名称と座標と表示するデータのデータ型を定義する画面定義情報に基づき、端末に画面表示のための情報を送信するオンライン型業務プログラムを容易に開発するための、端末メッセージマッピング支援システムがある。また、First Class誌、March/April, 1994, Vol. IV, Issue II のp.10, p.20に掲載されたWRAPPING LEGACY APPLICATIONSによれば、端末インタフェースを介して古いアプリケーションを制御するという概念が開示されている。

【0004】

【発明が解決しようとする課題】 従来のオンライン型業務プログラムは、ホスト計算機上で実行され、ホスト端末にオンライン型業務の画面表示を行うものであった。この画面は、文字を主体とするユーザインタフェースであり、例えば、複数の選択肢から一つを選ぶときは、各選択肢に対応付けられたコード番号をキーボードから入力しなければならなかった。また、画面表示の表現力に乏しく、情報を分かりやすく整理して表示したり、強調したい部分を強調の度合に応じて様々な大きさで表示したりすることが困難であった。これに対して、GUIを用いれば、例えば、上述の複数の選択肢からの選択操作では、選択したいものをマウスクリックすることで指示することができ、文字の大きさを何通りにも変化させることも、あるいは、枠で囲んだり、画像データを用いて絵を表示したりすることもできる。

【0005】 近年、パソコン（パーソナル・コンピュータ）などの小型の計算機が安価に入手可能となり、しかも、GUIにより操作性の向上と表現力の向上が図れるようになってきた。しかし、ホスト計算機上には、既に数十万本にも及ぶ極めて多数の業務プログラムがあり、それらの殆んどが文字を主体とするユーザインタフェースであるため、これらを改造することなくそのまま利用

でき、なお、かつ、上述のGUIの利点を活用するための技術が求められている。ホスト計算機上の業務プログラムと通信しながら動作し、かつ、GUIを提供するようなプログラムを新たに作成するためには、ホスト計算機上の業務プログラムと通信を行うプログラムと、GUIを提供するプログラムとが必要である。前述の従来技術によれば、ホスト計算機にアクセスするプログラムの作成を容易にするためのプログラム部品が提供されているため、ホスト計算機との通信プログラムの作成は容易になっているが、プログラム自体は、人手で作成しなければならなかった。

【0006】更に、ユーザがホスト計算機にアクセスするためのユーザインタフェースを提供するためには、ホスト計算機からデータを受信して適切な形式に変換した後これを表示する処理や、ユーザが入力したデータを適切な形式に変換した後これをホスト計算機に送信する処理も必要であるが、このような処理のプログラムも人手で作成しなければならなかった。しかも、オンライン型業務プログラムは、一般的に数十〜数百画面、多いものでは数千画面から構成されるにもかかわらず、その一方で、ホスト計算機とこれにアクセスするプログラムとの間の通信内容は、ホスト計算機上で実行されるオンライン型業務プログラムが表示する画面毎に異なり、同様に、ユーザインタフェースも画面毎に異なるため、画面毎にプログラムが必要になり、作成作業は多大な労力を必要としていた。このようなプログラムを自動的に作成できる技術は、従来存在しなかった。

【0007】また、前述の「EVB-SDK」という製品に見られる技術では、ユーザインタフェースの作成に便利なプログラム部品が提供されているが、これらのプログラム部品を配置したり、他のデータとのやりとりをコーディングしたりする手間がかかる。更に、ホスト計算機のデータの種類に応じて、利用するユーザインタフェースの部品を選択しなければならないが、複数の画面のそれぞれに対して適切な選択を行うことは困難であるという問題があった。更に、前述の「XMAP2E2」に示される技術では、ホスト計算機で実行する業務処理プログラムの作成において、画面定義データを利用することでプログラムの作成を支援していた。しかし、この業務処理プログラムにアクセスする方法ないし装置では、業務処理の内容によらない固定の機能を有する端末装置ないしは端末プロトコルに基づくアクセスに限定されていた。従って、業務の内容に応じて適切な表示を行うことができず、使いやすく美しいインタフェースの実現が困難となるという問題があった。

【0008】一方、ホスト計算機で実行する業務処理プログラムでは、計算やデータベースの更新などの業務処理の中核をなす処理以外に、画面入出力のためにホスト端末を制御する処理やエラーチェック処理などを行っていたため、プログラムの大きさが大きくなり、後から機

能を追加したり機能の変更をしたりするのが困難になっていた。また、前述のFirst Class誌の記事に見られる技術では、ホスト計算機と他の計算機とで互いに情報をやりとりする方法が示されていたが、この方法は従来の端末エミュレータなどに見られるものと同じであり、しかも、具体的なプログラムを作成する方法ないし装置は開示されていなかった。本発明は上記事情に鑑みてなされたもので、その目的とするところは、操作性の向上と豊かな表現力を用いた分かりやすい表示を行うためにGUIを用いてオンライン型業務処理を行うことを可能とするオンライン型業務処理システムを安価に提供することにある。

【0009】

【課題を解決するための手段】本発明の上記目的は、図1に示す如く、ホスト計算機となる第1の計算機(1)と、通信手段(3)とキーボードおよびディスプレイを備えかつグラフィックス表示機能を有する第2の計算機(2)とを有するオンライン型業務処理システムであって、前記第1の計算機(1)は、業務プログラムのソースコード(7)と、オンライン型業務処理画面における画面中のデータフィールドのフィールド名称、表示座標、少くとも入力か出力かを区別する入出力区分の対応表を定義する第1の画面定義情報(6)とを有し、前記業務プログラムのソースコード(7)と前記第1の画面定義情報(6)とから、該第1の画面定義情報(6)に従って前記通信手段(3)に接続された他の装置に画面表示を行って、前記業務プログラムのソースコード(7)に記述された処理手順に従ってオンライン業務処理を実行するオンライン業務プログラム(9)を生成する手段(8)を有し、前記第2の計算機(2)は、前記第1の画面定義情報(6)をコピーして得られる第2の画面定義情報(6)と、該第2の画面定義情報(6)から、GUIを表示して利用者との対話処理を行い、かつ、前記通信手段(3)を介して前記オンライン業務プログラム(9)との通信処理を行うGUI処理プログラム(40)を生成する手段(30)を有することを特徴とするオンライン型業務処理システムによって達成される。

【0010】

【発明の実施の形態】本発明に係るオンライン型業務処理システムにおいては、従来から、オンライン業務プログラムがホスト端末装置と通信するデータの形式が、ホスト端末装置の仕様によって決められているため、多くのオンライン業務プログラムに共通の形式であることに着目し、この形式を合わせるという手段により、既存のオンライン業務プログラムを修正せずにそのまま利用して、これと通信して動作するホストアクセスプログラムをGUI処理プログラムとして作成することを可能としたものである。しかも、このGUI処理プログラムは、ホスト計算機となる第1の計算機の画面の内容を定義する画面定義情報を用いて自動的に生成するという手段に

より、プログラム開発の労力を削減し、安価に作成することができる。すなわち、本発明に係るオンライン型業務処理システムにおいては、文字や罫線を主体とした従来の使いにくい画面インタフェースを備えたホスト計算機上の既存のオンライン型業務プログラムを修正することなくそのまま利用しつつ、このプログラムとの通信およびGUIの制御を行うGUI処理プログラムを自動的に作成するものである。

【0011】また、ホストアクセスプログラムであるGUI処理プログラムが既存のオンライン業務プログラムと通信するためには、上述のホスト端末装置の仕様に従った形式のデータとプログラム内部の形式との相互の変換が必要であるが、本発明に係るオンライン型業務処理システムにおいては、画面定義情報に含まれているデータの名称と、そのデータが従来のホスト端末装置に表示されるとき座標と座標の情報を、通信データに含まれる座標情報と照合することで相互の変換を可能としている。更に、GUI処理プログラムはGUIを画面表示したり、ユーザからの指示を受け付けたりする処理が必要であるが、本発明に係るオンライン型業務処理システムにおいては、画面定義情報に含まれているデータの名称とデータの座標の情報をを用いることで、GUIを画面表示するための座標を決定し、更に、画面定義情報に含まれているデータの名称と、そのデータが利用者が入力するデータなのか利用者に表示するデータなのかという区分との情報を用いることで、どのような種類のGUIを用いるかを決定し、これらの処理を行うためのプログラム、つまりGUI処理プログラムを自動的に生成する。

【0012】更に、前述のXMAP2E2に示される従来技術を用いてオンライン型業務プログラムを開発した場合、ホスト画面の内容を定義する画面定義情報が既に作成されているため、これを用いることにより、画面定義情報を作成する労力を削減する。また、従来のホスト端末装置の代わりにパソコンやワークステーションなどのグラフィックス表示が可能な計算機をホスト計算機と通信手段を介して接続する構成とし、かつ、パソコンやワークステーション上でホストアクセスプログラムであるGUI処理プログラムを実行することにより、オンライン型業務プログラムを変更しなくても、ホストアクセスプログラムであるGUI処理プログラムの方を変更するだけでエラーチェック処理などの新たな機能を追加したり、入力方法の変更などの機能変更をしたりできるため、修正箇所が限定されるために変更が容易となる。更に、パソコンやワークステーションのグラフィックス表示機能を活用して、例えば、業務処理の内容を説明したり入力方法を説明したりする表示を、美しく表示するように変更することもできる。

【0013】以下、本発明の実施例を図面に基づいてより詳細に説明する。最初に、本発明の概要を、図1～4を参照して説明する。図3は、従来の一般的なオンライ

ン業務プログラムが端末装置に表示する画面の一例を示している。また、図4は、本発明に係るオンライン型業務処理システムにより作成できるホストアクセスプログラムであるGUI処理プログラムの画面表示例を示しており、パソコンやワークステーション計算機に搭載されたウィンドウシステムの機能を活用することで、使いやすくしかも表示が美しいGUIを提供することができる。前述の如く、本発明の特徴は、既存のオンライン業務プログラムを変更することなく、オンライン型業務処理装置をより使いやすく、より美しい表示にすることを可能とする点にある。

【0014】まず、従来のオンライン型業務処理装置について説明する。図2は、従来のオンライン型業務処理装置の構成例を示すブロック図である。図2において、1はホスト計算機である。ホスト計算機1は、汎用計算機とかメインフレーム計算機と呼ばれることもあるが、本明細書中では「ホスト計算機」または単に「ホスト」と記載する。14はディスプレイとキーボードを有し、画面表示したり利用者の入力を受付けるホスト端末装置である。3はホスト計算機1とホスト端末装置14との間の通信に用いる通信手段である。ホスト計算機1において、6は画面定義情報である。画面定義情報6はオンライン業務プログラム9の生成に用いる。7は業務プログラムソースであり、オンライン業務処理の処理手順をプログラミング言語で記述したものである。

【0015】また、ホスト計算機1において、8はオンライン業務プログラム生成手段であり、画面定義情報6に従ってホスト端末装置14に定義通りの画面を表示する処理を生成すると共に、業務プログラムソース7をコンパイルしてオンライン業務処理を生成し、更に、これらを組み合わせることでオンライン業務プログラム9を生成する。オンライン業務プログラム9はこのようにして生成されたものであり、これを実行すると、通信制御装置4および通信手段3を経由して、通信回線に接続されたホスト端末装置14にオンライン業務画面を表示する。利用者はこの画面を操作して業務を遂行する。

【0016】次に、本発明に係るオンライン型業務処理システムについて説明する。本発明に係るオンライン型業務処理システムでは、前述の如く、従来のオンライン型業務処理装置で作成済みのオンライン業務プログラム9を変更せずにそのまま利用することで、開発を容易としている。このために画面定義情報6を利用する点について、以下に説明する。図1は、本発明に係るオンライン型業務処理システムの構成を示すブロック図である。図1において、1はホスト計算機、2は計算機、3はホスト計算機1と計算機2との間で画面表示データおよび画面入力データを相互に通信するための通信手段である。計算機2としては、パソコンやワークステーション計算機などを用いることができる。

【0017】ホスト計算機1に含まれる構成要素は、既

に説明した図2のホスト計算機1と同じである。計算機2に含まれる構成要素について、以下、説明する。6はホスト計算機1に含まれる画面定義情報を計算機2にコピーしたものである。図7に基づいて、画面定義情報6の詳細について説明する。90は定義している画面の名称を示す。91は画面中に含まれるフィールドの名称を示す。これら90と91とがキーであり、図7の各々の行は、ある画面中のあるフィールド1個についての定義を示している。92はそのフィールドの入出力区分を示す。93と94はそのフィールドを画面中のどこに配置するかを示す座標である。95はそのフィールドの長さであり、その長さ分の文字列の表示や入力ができる。96はそのフィールドで扱うデータの型を示す。97はフィールドの色や文字の大きさなどの属性情報を示し、ここではそれぞれのフィールドの色を定義した例を示している。98は入出力区分が固定のときだけ使われるもので、固定的に表示する文字列を示す。

【0018】図1に戻って、他の構成要素について説明する。13は処理文テンプレートであり、ホストアクセスプログラムであるGUI処理プログラム40の処理手順を各部分ごとに記述したものである。この処理文テンプレート13を画面定義情報6から得られる情報に従って組み合わせたり、処理文テンプレート13中の可変部分に画面定義情報6から得られる座標値などの具体的な値を設定したりすることで、GUI処理プログラム40が得られる。このようにしてGUI処理プログラム40を生成するのが、GUI処理プログラム生成手段30である。GUI処理プログラム生成手段30では、表示情報設定処理生成手段32において表示情報設定処理19を、入力情報取得処理生成手段33において入力情報取得処理20を、また、これらを組み合わせてサブルーチン生成手段35においてサブルーチン18を各画面ごとに生成し、また、表示画面制御処理生成手段31において表示画面制御処理22を生成し、これと全体の初期化処理とを組み合わせるメインルーチン17を生成する。

【0019】表示情報設定処理生成手段32は、表示情報設定処理に対応する処理文テンプレート13を用いて、表示情報設定処理19を生成する。処理文テンプレート13には、画面表示情報21から得られる情報に基づいて画面表示を行うための定型的な処理が含まれているが、画面表示情報21は画面内容に応じて変化するため、これを解析して適切な表示をするためには画面定義情報6が必要である。このため、表示情報設定処理生成手段32は、画面定義情報6から得られる入出力区分に従って、画面表示に用いられるフィールドか否かを識別し、座標、長さ、属性に従って、指定の位置・長さに指定の属性で画面表示を行う処理を生成する。また、固定的にある文字列を表示するフィールドでは、画面定義情報6の文字列の指定に従って表示する処理を生成する。

【0020】また、GUI処理プログラム40では、G

UI部品ライブラリ15を用いてGUIを表示する。GUI部品ライブラリ15を呼び出す処理は、処理文テンプレート13に含まれており、従って、表示情報設定処理生成手段32が生成する表示情報設定処理19がGUI部品ライブラリ15を呼び出す。例えば、固定的にある文字列を表示するときは、入力機能を持たないGUI部品を用いて表示し、また、文字の入力と出力の両方が必要な処理については、文字の入力編集機能を持つGUI部品を用いて表示する処理を処理文テンプレート13に指定することで、GUI部品ライブラリ15を呼び出す処理が生成される。

【0021】同様に、入力情報取得処理生成手段33は、入力情報取得処理20に対応する処理文テンプレート13を用いて、入力情報取得処理20を生成する。画面内容に応じて変化する部分については、画面定義情報6から得られる情報に従って生成する。入力情報取得処理20は、表示情報設定処理19が表示した画面に対して、GUI処理プログラム40の利用者が画面入力したとき、入力された情報を取得する処理である。

【0022】サブルーチン生成手段35は、サブルーチンに対応する処理文テンプレート13を用いてサブルーチン18を生成する。サブルーチン18の中で、表示情報設定処理19の部分は、表示情報設定処理生成手段32にて生成したものを処理文テンプレートに埋め込んで生成し、また、入力情報取得処理20の部分は、入力情報取得処理生成手段33にて生成したものを処理文テンプレートに埋め込んで生成する。画面定義情報6には複数の画面に関する定義を含み、サブルーチン18は画面ごとに一つずつに必要であるから、GUI処理プログラム生成手段30ではサブルーチン生成手段35を繰り返し用いて、すべての画面についてサブルーチンを生成する。

【0023】表示画面制御処理22は、画面表示情報6に含まれる画面名に従って、その画面に対応するサブルーチン18を呼び出す処理である。表示画面制御処理生成手段31は、表示画面制御処理22の処理文テンプレートに、画面表示情報6から得られる画面名を用いて画面を識別する処理を埋め込むことで、表示画面制御処理22を生成する。メインルーチン生成手段34は、メインルーチン17の処理文テンプレート13を用いて全体初期化処理を生成し、これに表示情報設定処理生成手段32が生成した表示画面制御処理22を組み合わせるメインルーチン17を生成する。

【0024】以下、図1を参照しながら、GUI処理プログラム40の動作について、説明する。メインルーチン17では、通信制御装置5を制御してオンライン業務プログラム9からの通信データを受信し、画面表示情報21を得る。データの受信処理は通信処理ライブラリ12が行う。通信処理ライブラリ12を呼び出す処理は、メインルーチン生成手段34が用いたメインルーチンの

処理文テンプレートに含まれる。続いて、メインルーチン17では、得られた画面表示情報21を、表示画面制御処理22を用いて、画面名に応じてその画面に対応するサブルーチン17を呼び出す。

【0025】そのサブルーチン中の表示情報設定処理19が画面を表示する。表示された画面に対する利用者の入力指示を入力情報取得処理20が処理し、オンライン業務プログラム9に送信すべき画面表示情報を作成する。メインルーチン17では、この画面表示情報をオンライン業務プログラム9に送信する。オンライン業務プログラム9への送信処理は、通信処理ライブラリ12が行う。通信処理ライブラリ12を呼び出す処理はメインルーチン生成手段34が用いたメインルーチンの処理文テンプレートに含まれる。以後、これらの処理を繰り返すことで、オンライン業務プログラム9とのデータの送受信を繰り返しながら、画面表示と、利用者からの入力指示の処理を繰り返す。

【0026】図2に示した従来のオンライン型業務処理装置の構成では、ホスト端末装置14がオンライン業務プログラム9とデータの送受信を行いながら画面表示とユーザの入力指示の受け付けをしていたが、図1の構成では、通信処理ライブラリ12を用いることでホスト端末装置14の動作を模擬することにより、元来はホスト端末装置14と通信して動作するように作成されたオンライン業務プログラム9とのデータ送受信を可能とし、しかも、画面定義情報によって生成された表示情報設定処理19がGUI部品ライブラリ15を用いてGUI画面を表示することで、GUIを提供する構成としている。

【0027】以下、図1～13を用いて、本発明の具体的な実施例を詳細に説明する。まず、本実施例で用いるオンライン型業務例について、簡単に説明する。図3は会社の出張旅費を必要経費として清算するため、出張者本人が旅費を申告する、従来の一般的なオンライン型業務画面例を示す。図3において、80の部分は常に「氏名」と表示する固定領域である。81は出張者の氏名を表示する出力領域である。82は経費の分類のために必要な番号を出張者本人が入力するための入出力領域を示す。但し、この番号は特別な出張の場合を除き、出張者によって固定的に決まっているため、予めその番号を表示する。特別な出張の場合にのみ、出張者が番号を入力する。87は旅費の金額を入力するための入力領域である。出張者はこの画面に必要な事項を入力して、入力完了を示すキーボード入力を行う。

【0028】オンライン業務プログラム9は、上述の入力データに従って、旅費精算処理を実行する。以下、本実施例の処理手順を説明する。図5は、本実施例に係るオンライン型業務処理システムにおける処理手順を示すフローチャートである。図5において、まず、画面定義情報6に従ってオンライン業務プログラム9を生成する(ステップ51)。画面定義情報の一例を、図7に示す。

画面定義情報は画面の構成要素に関する座標やデータ型などの情報を有し、図3に示す如きオンライン型業務の画面中の個々の要素に対応している。

【0029】次に、画面定義情報に従って、ホストアクセスプログラムであるGUI処理プログラム40を生成する(ステップ52)。GUI処理プログラム40は1個のメインルーチン17と、1個または複数のサブルーチン18とからなり、それぞれのサブルーチンは1つのオンライン業務画面に対応している。上述のステップ51、ステップ52の処理と、以下に述べるステップ53以下の処理とは、処理を行う時刻に隔たりがある。その意味で、ステップ51、ステップ52およびステップ53以下に、191、192、193という符号を付与している。

【0030】つまり、本実施例に係るオンライン型業務処理システムは、ホスト計算機上の既存のオンライン業務プログラムを修正することなく、パソコンなどの安価な計算機から継続利用するためのものである。通常、オンライン業務プログラムは図2に示した構成で利用するために作成され、しばらくは、図2に示した構成で運用された後に、本発明に係る処理手順が適用される(191)。従って、例えば、191でのオンライン業務プログラムの作成は10年前のことであり、192でのGUI処理プログラム40の生成が現時点でのことで、今後はステップ53以下(193)の処理を行い、GUIを用いて業務処理を行うということが行われる。

【0031】以下、193のステップ53以下の処理手順を説明する。まず、ホスト計算機中のオンライン業務プログラム9と、ステップ52で生成したGUI処理プログラムを起動する(ステップ53)。次に、オンライン業務プログラム9がGUI処理プログラム40へ画面表示情報を送信する(ステップ54)。オンライン業務プログラム9は、これを21に記憶する。オンライン業務プログラム9は、図2のような従来の構成で動作していたものを、変更せずにそのまま動作させている。このため、オンライン業務プログラム9が、図1中の通信手段3を介して送信する画面表示情報は、図2の構成のときと同じものである。

【0032】図8に、画面表示情報の一例を示す。図8において、130は画面名を表わす。131の「F」と表記した部分は、固定フィールドに関するデータが後に続くことを表わす。132と133とは、それぞれ、固定フィールドのX座標、Y座標を示す。134は固定的に表示する文字列である。以後、同様に、フィールドの種類を表わすコードとX座標、Y座標、文字列のデータが続く。なお、135は出力フィールド、136は入出力フィールド、137は入力フィールドを表わしている。入力フィールドは文字列データはなく、座標だけである。

【0033】次に、画面表示情報の画面名に対応する表

示情報設定処理19を起動する(ステップ55)。表示情報設定処理19では、画面表示情報21の残りの部分の情報に従って、図4に示すようなGUIを表示する(ステップ56)。ステップ57では、ユーザ入力を受け付ける。このステップ57は、図1中のGUI部品ライブラリ15で実行する。利用者がデータを入力すると、GUI部品ライブラリ15は、現在処理中のオンライン業務画面に対応する入力情報取得処理20を起動する。

【0034】ステップ58では、入力情報取得処理20において、入力情報を画面表示情報の形式に変換してオンライン業務プログラム9へ送信する。次に、ステップ59で、オンライン業務プログラムにおいて、入力情報を処理する。以上に説明した処理手順によれば、画面定義情報に従ってオンラインプログラム生成手段8が生成した、既存のオンライン業務プログラム9を変更することなく、図1の構成の中で実行する。

【0035】GUI処理プログラム生成手段30は、画面表示情報21と図4のようなGUIとの情報の設定と取得の処理を、画面定義情報を用いて生成し、これを実行する。従って、この方法により、ホスト計算機上の既存のオンライン業務プログラムを有効に活用しながら、かつ、図4のような使いやすく美しいGUIを表示することが可能となる。以下、図6を参照しながらステップ52の処理手順を詳細に説明する。図6は本発明に係るGUI処理プログラム生成処理の処理手順を示すフローチャートである。

【0036】図6において、まず、変数Kに空文字列を代入する(ステップ61)。変数Kは後のステップ75において、表示画面制御処理22の生成に用いる。表示画面制御処理22は、ステップ55を実行する部分である。次に、画面定義情報ごとの繰り返し処理により、1個または複数のオンライン業務画面を各画面単位に処理する(ステップ62)。次に、画面定義情報を1画面分取得する(ステップ63)。画面定義情報の一例を図7に示す。図7において、100は図3の80に対応する定義を示し、100の中での文字列98の項目によれば、図3の80のように表示される文字列「氏名」を定義している。

【0037】図3と図7を参照しながら、図7の他の項目について以下説明する。90は定義している画面の名称を示す。ここでは1画面分について処理中なのですべて同じ名称である。91は画面中に含まれるフィールドの名称を示す。図7の各々の行は、ある画面中のあるフィールド1個についての定義を示している。92はそのフィールドの入出力区分を示し、固定フィールドはある固定の文字列を常に表示するものであり、出力フィールドはホスト計算機から送られる情報を表示するものであり、入力フィールドは利用者が入力したデータをホストに送るためのものであり、入出力フィールドは最初にホストから送られる情報を表示し、かつ、利用者が入力し

たデータをホスト計算機に送るためのものである。

【0038】93と94はそのフィールドを画面中のどこに配置するかを示す座標である。95はそのフィールドの長さであり、その長さ分の文字列の表示や入力を行うことができる。96はそのフィールドで扱うデータの型を示し、日本語は日本語の文字列であり、英数はアルファベットと数字と記号文字からなる文字列であり、数字は数字文字列である。97はフィールドの色や文字の大きさなどの属性情報を示し、ここではそれぞれのフィールドの色を定義した例を示している。98は入出力区分が固定のときだけ使われるもので、固定的に表示する文字列を示す。なお、画面名、フィールド名には任意の文字列を用いることができるが、一般的には名称規則を決めることが多い。

【0039】例えば、この場合、最初の「AB」がオンライン業務処理全体に共通な識別コードであり、次の「P」ないしは「F」は、それぞれ画面名、フィールド名を表わす。フィールド名の4文字目は入出力区分に応じて、Fは固定、Oは出力などを表わす。残りの部分は、各フィールドごとに異なる名称を表わす。図6において、次に、変数Pに画面名を代入する(ステップ64)。次に変数I、Sに空文字列を代入する(ステップ66)。変数Iは後のステップ73において表示情報設定処理19の生成に用い、変数Sは同じくステップ73において入力情報取得処理20の生成に用いる。

【0040】次に、画面定義情報中のフィールド1個分の定義について、繰り返し処理する(ステップ67)。次に、フィールド1個分の情報を取得して、変数Fにフィールド名を、変数FKに入出力区分を、変数FXにX座標を、変数FYにY座標を、変数FLに長さを、変数FAに属性を、変数FSに文字列をそれぞれ代入する(ステップ68)。例えば、図7の100によれば、変数Fに「ABFFNAME」という文字列を代入する。

【0041】次に、処理文テンプレート13からフィールド設定処理のテンプレートを取得して、テンプレート中の変数を変数の値に置換することでフィールド設定処理を生成し、結果を変数Iに追加する(ステップ69)。フィールド設定処理のテンプレートは、フィールドの入出力区分に応じて別々のものを容易しており、変数FKの値に対応するテンプレートを取得する。図10に、固定フィールドに用いるフィールド設定処理のテンプレートの一例を示す。図10において、「\$」記号がある部分が変数を示す。例えば、151の部分では、「\$F」の部分を変数Fの値「ABFFNAME」に置き換えることで、「f_ABFFNAME」という文字列になる。

【0042】処理結果を、図9の143に示す。143は図4において、110の「氏名」という文字列を表示するためのプログラムである。変数Iはステップ69の処理結果を、すべてのフィールドの定義について蓄積してゆく。次に、ステップ69と同様に、フィールド取得処

理のテンプレートをを用いてフィールド取得処理を生成し、結果を変数Sに追加する(ステップ71)。以上の繰り返しにより、すべてのフィールドの定義についての処理結果が変数I、Sに蓄積されたので、次に、処理文テンプレート13からサブルーチンのテンプレートを取得し、同様に変数の置換を行うことでサブルーチンを生成し、記憶する(ステップ73)。

【0043】図11に、サブルーチンのテンプレートの一例を示す。152の部分にフィールド設定処理を置換し、153の部分にフィールド取得処理を置換する。サブルーチンの生成結果を、図9に示す。図9において、141の部分にフィールド設定処理とその他の画面初期化処理からなる表示情報設定処理19の一例を示す。また、142の部分はフィールド取得処理他からなる入力情報取得処理20の一例を示す。次に、処理文テンプレート13から表示画面制御処理のテンプレートを取得して、変数Pの置換を行うことで表示画面制御処理を生成し、これを変数Kに追加する(ステップ74)。ここでは、変数Pに記憶した画面名は「ABP010」であるから、この画面に関する表示画面制御処理を追加する。図12において、154の部分が今回変数Kに追加する部分である。

【0044】以上の繰り返しにより、すべての画面についての処理結果が変数Kに得られるので、次に、これをメインルーチンのテンプレートに適用してメインルーチンを生成し、記憶する(ステップ75)。図12に、生成したメインルーチンの一例を示す。以上、説明した処理手順によれば、画面定義情報を用いて、オンライン業務プログラムと互いに通信しながら動作するGUI処理プログラムを自動的に生成する。このプログラムを、GUI部品ライブラリ15と、通信制御装置5を用いて通信するための通信処理ライブラリ12で、ともにコンパイルして実行モジュールを作成し実行することで、ステップ54～59を処理する。

【0045】生成したプログラムの動作を、図9と図12を参照しながら説明する。図12において、まず、初期化処理およびホスト計算機との通信回線の接続を行う(ステップ161)。この処理は、通信処理ライブラリ12を呼び出すことで行う。次に、オンライン業務プログラムから画面表示情報を受信し、変数msgに代入する(ステップ162)。このステップはステップ54において、GUI処理プログラム側の受信処理である。次に、「formSet」において、画面表示情報から画面名を取得する(ステップ163)。

【0046】図8によれば、130の部分を取得する。次に画面名に対応する表示情報設定処理を起動する。130によれば画面名は「ABP010」であるから154の部分の処理により「w_ABP010_Set」を起動する。次に、図9において、画面表示情報をメモリ中での記憶構造に変換して記憶する(ステップ164)。図13

に、画面表示情報の記憶構造を示す。図13において、各領域は1バイトのデータに対応し、全体では2次元配列を構成している。

【0047】以下、説明の都合上、この配列変数の名称を「G」とすると、右方向の順位を第一の添字、下方向の順位を第二の添字とすることで、例えば、180の部分はG(0, 0)と表現できる。同様に、181はG(1, 0)と表現でき、182はG(0, 1)と表現できる。この2次元配列変数Gの要素を、図3に示す従来のオンライン業務画面と対応づけて記憶することにより、オンライン業務プログラムとの通信に用いるデータ形式と、内部での形式との相互の変換を行う。例えば、図8の131～134によれば、文字列「氏名」は座標(2, 2)に表示されるべきもので、(漢字文字は1文字につき2バイト必要なので)これをメモリ中では、183のG(2, 2)、G(2, 3)、G(2, 4)、G(2, 5)の4バイトのメモリ領域に記憶する。

【0048】次に、GUIを表示するウィンドウを初期化する(ステップ165)。次に、図7の100に定義された固定文字列を表示するための表示GUI部品を初期化する(ステップ166)。関数「labelCreate」は表示GUI部品の初期化を行う。次に、固定的に表示すべき文字列「氏名」を設定する(ステップ167)。次に、97の属性に応じて表示色を設定する(ステップ168)。以上、ステップ166～168によりフィールド1個分に対応するGUIを設定する。これにより、図4の110のGUIを表示する。以下、他のフィールドについて同様の処理を行う。但し、101に対応する出力フィールドでは、表示する文字列は毎回変化するので、オンライン業務プログラムから受信した画面表示情報を、図13のような変数Gを参照しながら取得する。

【0049】ここでは、184のG(12, 2)以降に格納している「鈴木 一」を取得して設定する(ステップ169)。関数「messageGet」は、第一引数にX座標、第二引数にY座標、第三引数に取得する文字列の長さを指定することで、変数Gを用いて、オンライン業務プログラム9から受信した画面表示情報の中の、当該座標部分の文字列を取得する。また、102に対応する入出力フィールドでは、画面表示とユーザ入力の間が必要なので、入出力GUIを初期化する(ステップ170)。関数「textCreate」は入出力GUI部品の初期化を行う。また、103に対応する入力フィールドでは入力だけなので表示情報の設定は行わない。

【0050】次に、ユーザが入力終了を指示するためのボタンGUIを初期化し、ボタン押下時に、処理「w_ABP010_Get」を起動するように設定する(ステップ171)。以上の処理により、図4に示すGUI画面を表示する。次に、GUI部品がユーザからの入力を受付、ボタン押下時に処理「w_ABP010_Get」を起動する。次に、図4中の利用者が112に入力した振

替番号を取得する(ステップ172)。関数「textGetCaption」は、入出力GUIに入力された文字列を取得する。

【0051】次に、これを変数Gに設定する(ステップ173)。関数「messageSet」は、第一引数に設定する文字列、第二引数にX座標、第三引数にY座標、第四引数に設定する文字列の長さを指定することで、変数Gの当該座標部分に対応する要素に文字列を設定する。以下、他の入力フィールド、入出力フィールドについて同様に、変数Gに設定する。次に、変数Gから画面表示情報を取得して、オンライン業務プログラムに送信する(ステップ174)。このステップはステップ58において、ホストアクセスプログラムの側の送信処理である。次に、GUI画面を消去し、ステップ162以下の処理を継続する(ステップ175)。

【0052】以上、説明した処理手順によれば、図2に示す従来のオンライン型業務処理装置において使用していたオンライン業務プログラム9を、修正することなく、かつ、図4に示す使いやすく美しいGUIにより操作可能な、オンライン型業務処理を得る。なお、上記実施例は本発明の一例を示したものであり、本発明はこれに限定されるべきものではないことは言うまでもないことである。例えば、オンライン業務プログラム9はホスト計算機以外の計算機で実行することも可能で、例えばオンライン業務プログラムをワークステーション計算機上の言語コンパイラで再コンパイルしてワークステーション計算機上で実行することもできる。この場合でも通信手段3を経由して通信するデータの内容は、例え通信プロトコルに違いはあっても意味的には等価のものであり、従って本発明を同様に適用することは容易である。

【0053】

【発明の効果】以上、詳細に説明した如く、本発明によれば、ホスト計算機上の既存のオンライン業務プログラムの修正も追加のプログラム記述もなしに、このプログラムを用いたオンライン業務の画面表示を、使いやすく美しいGUIにすることができるといふ顕著な効果を奏するものである。表示のための装置として、従来のホスト端末装置の代わりにパソコンやワークステーションなどの、グラフィックス表示が可能な計算機が必要となるが、これらは近年非常に安価に入手できる。

【0054】特に、画面表示をGUIにするホストアクセスプログラムは、オンライン業務プログラムを作成するために既に作成されている画面定義情報を利用するため、画面定義のための労力を必要とせず、自動的に作成する。しかもこの、画面定義情報を用いてホストアクセスプログラムを自動的に作成するプログラムは、ホスト計算機上の様々な既存のオンライン業務プログラムに共通して適用できるため、一度これを作成すれば他のオンライン業務プログラムにも利用できる。また、自動的に作成したホストアクセスプログラムを更に修正して、利

用者の入力ミスに対するエラーチェック処理を追加することもできる。これにより、ホスト計算機との通信をすることなく入力誤りをチェックするため、レスポンスが向上する。しかもエラーチェック処理のためにホスト計算機に負荷をかけずに済む。

【図面の簡単な説明】

【図1】本発明の一実施例に係るオンライン型業務処理システムの構成を示すブロック図である。

【図2】従来のオンライン型業務処理装置の構成例を示すブロック図である。

【図3】従来の一般的なオンライン業務プログラムが端末装置に表示する画面の一例を示す図である。

【図4】実施例に係るオンライン型業務処理システムにより作成できるGUI処理プログラムの画面表示例を示す図である。

【図5】実施例に係るオンライン型業務処理システムにおける処理手順を示すフローチャートである。

【図6】実施例に係るホストアクセスプログラム生成処理(ステップ52)の処理手順を示すフローチャートである。

【図7】実施例に係る画面定義情報6の一例を示す図である。

【図8】実施例に係る、通信手段3を介して通信する画面表示情報の一例を示す図である。

【図9】実施例に係る、ステップ52で生成されるホストアクセスプログラムにおけるサブルーチン部分の一例を示す図である。

【図10】図9のプログラムの生成に用いる、処理文テンプレート13の一例を示す図である。

【図11】図9のプログラムの生成に用いる、処理文テンプレート13の他の例を示す図である。

【図12】ステップ52で生成されるホストアクセスプログラムにおけるメインルーチン部分の一例を示す図である。

【図13】画面表示情報21の記憶構造を示す図である。

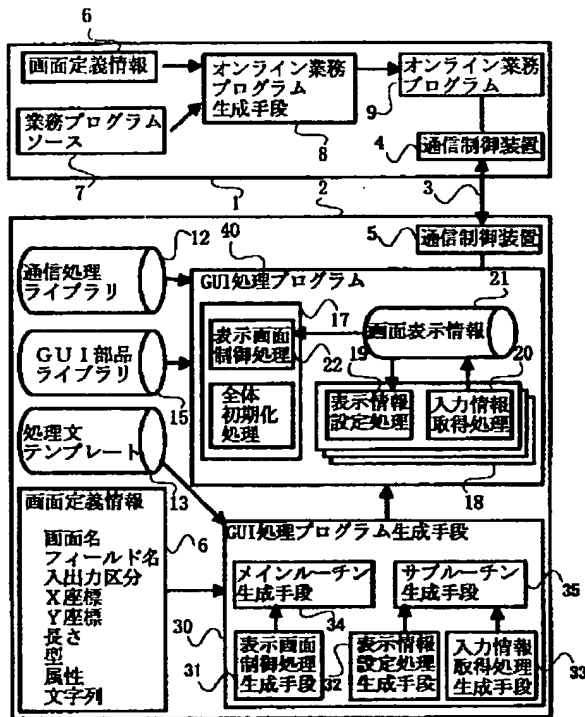
【符号の説明】

- 1 ホスト計算機
- 2 計算機
- 3 通信手段
- 4 通信制御装置
- 5 通信制御装置
- 6 画面定義情報
- 9 オンライン業務プログラム
- 12 通信処理ライブラリ
- 13 処理文テンプレート
- 15 GUI部品ライブラリ
- 17 メインルーチン
- 18 サブルーチン
- 19 表示情報設定処理

- 20 入力情報取得処理
21 画面表示情報
22 表示画面制御処理

- 30 GUI処理プログラム生成手段
40 GUI処理プログラム

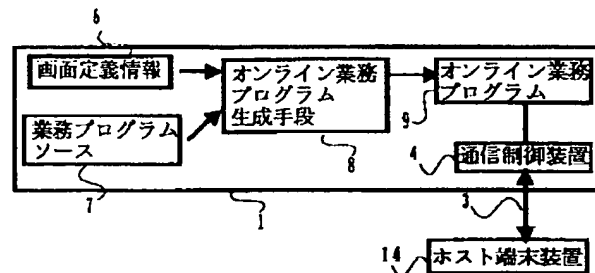
【図1】



【図3】

【図8】

【図2】

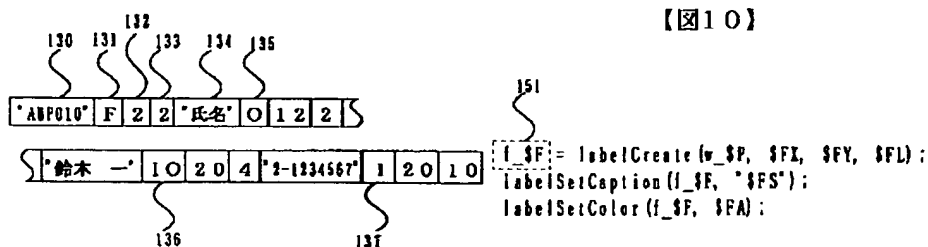


【図7】

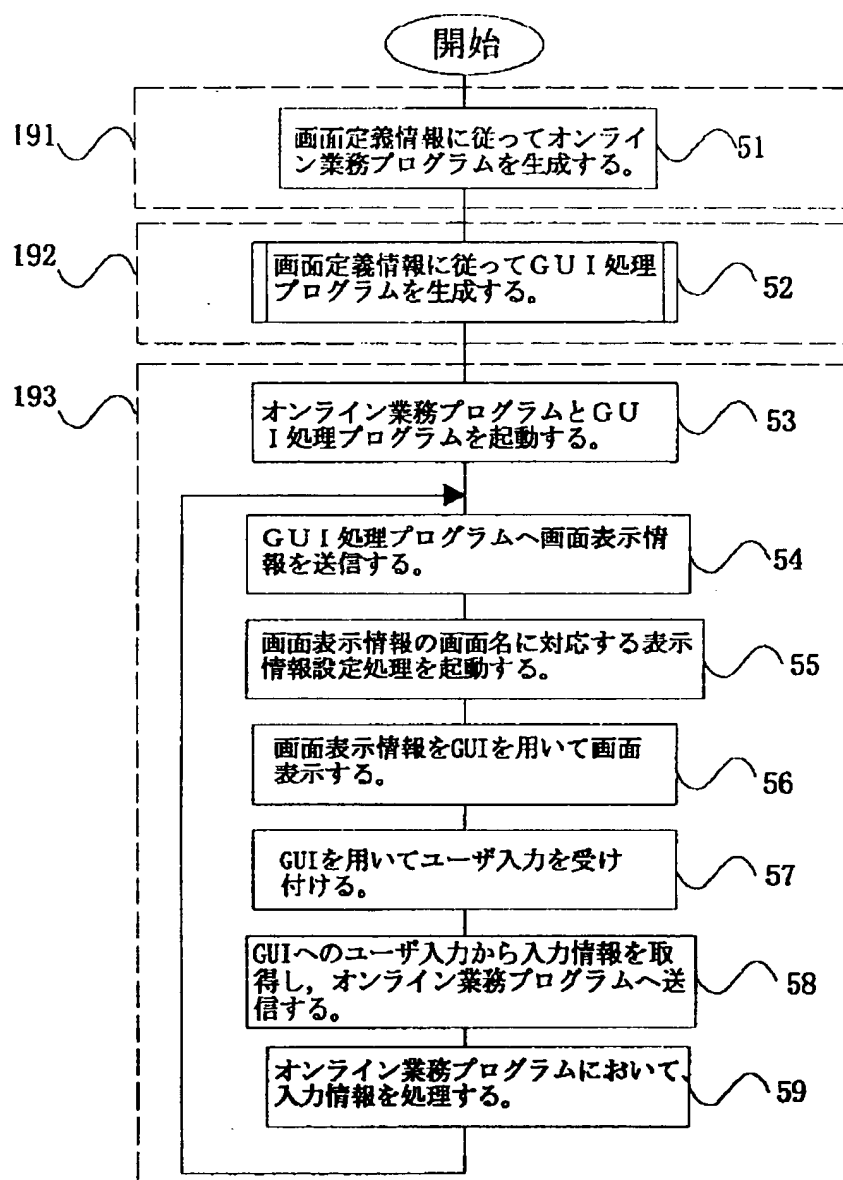
	90	91	92	93	94	95	96	97	98
	5	5	5	5	5	5	5	5	5
	画面名	フィールド名	入出力区分	X座標	Y座標	長さ	型	属性	文字列
100~	ABP010	ABPFNAME	固定	2	2	4	日本語	青	氏名
101~	ABP010	ABFONAME	出力	12	2	12	日本語	白	
102~	ABP010	ABFEFURI	入出力	20	4	10	英数	緑	
103~	ABP010	ABPIRYOB	入力	20	6	7	数字	緑	

【図4】

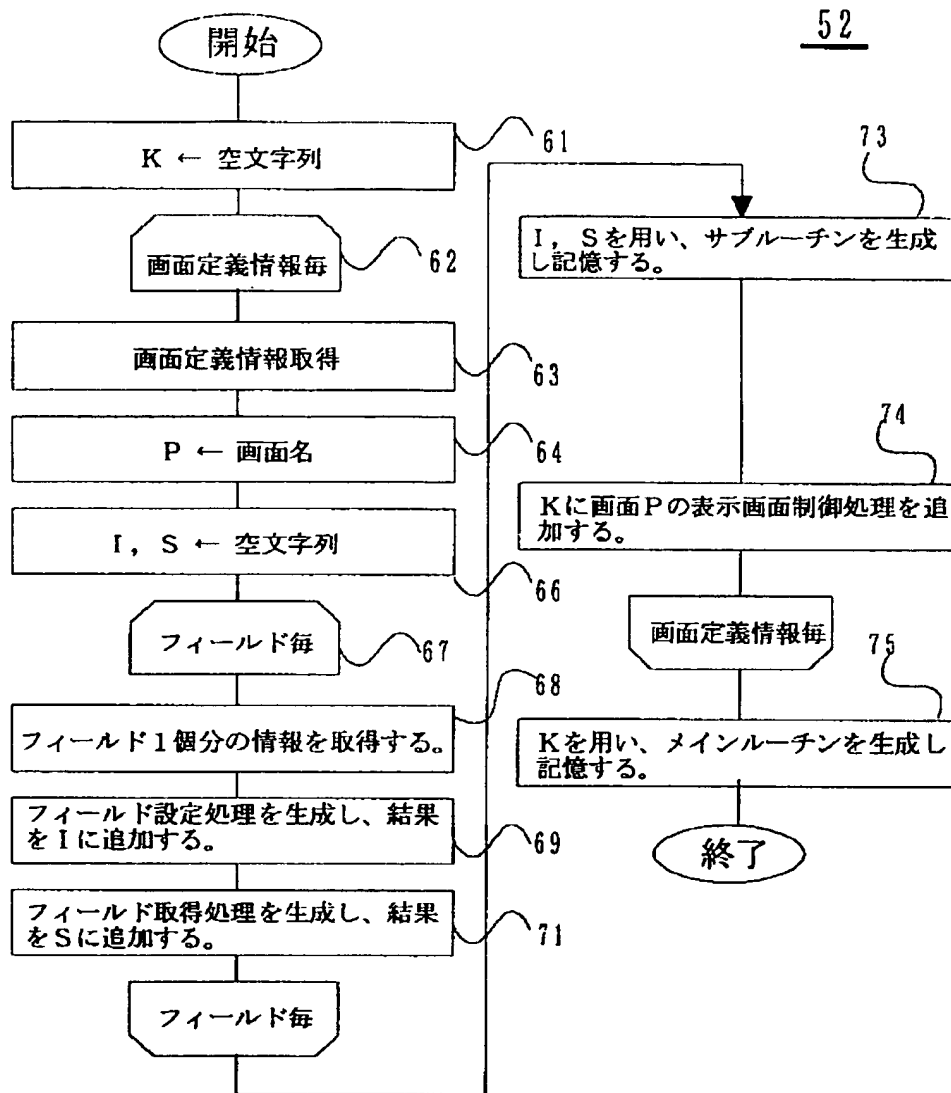
【図10】



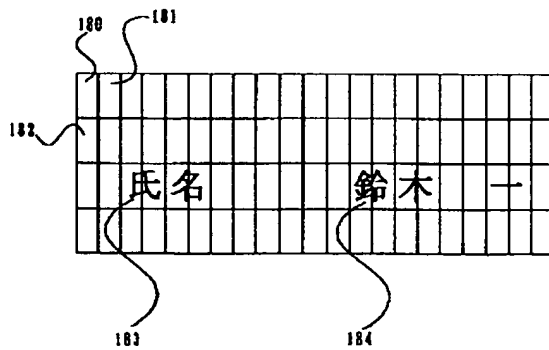
【図5】



【図6】



【図13】



【図9】

```

String s;
String messageGet(int x, int y, int len);
String messageSet(String msg, int x, int y, int len);
void w_ABPO10_Set(String msg) {
164 messageMap(msg);
    w_ABPO10 = windowCreate 0; 165
166 {
    ABFFNAME = labelCreate(w_ABPO10, 2, 2, 4);
    labelSetCaption(l_ABFFNAME, "IC名");
168 labelSetColor(l_ABFFNAME, COLOR_BLUE); 167
    ABFONAME = labelCreate(w_ABPO10, 12, 2, 12);
169 s = messageGet(12, 2, 12);
    labelSetCaption(l_ABFONAME, s);
    labelSetColor(l_ABFONAME, COLOR_WHITE);
170 ABFEFURI = textCreate(w_ABPO10, 20, 4, 10);
    s = messageGet(20, 4, 10);
    labelSetCaption(l_ABFEFURI, s);
    textSetColor(l_ABFEFURI, COLOR_GREEN);
    ABFIRYOH = textCreate(w_ABPO10, 20, 6, 7);
    textSetColor(l_ABFIRYOH, COLOR_GREEN);
171 b_SEND = buttonCreate(w_ABPO10, BUTTON_LOWER_RIGHT,
    "実行");
    buttonAddEvent(b_SEND, BUTTON_CLICK, w_ABPO10_Get);
}

void w_ABPO10_Get 0 {
    String msg;
173 s = textGetCaption(l_ABFEFURI); 172
    messageSet(s, 20, 4, 10);
    s = textGetCaption(l_ABFIRYOH);
    messageSet(s, 20, 6, 7);
    msg = messageUnmap 0; 174
    hostSend(msg);
    windowDelete(w_ABPO10); 175
    msg = hostReceive 0;
    formSet(msg);
}

```

【図12】

```

String s;
void main 0 {
    String msg;
    initialise 0;
    hostConnect 0; 161
    msg = hostReceive 0; 162
    formSet(msg);
}

void formSet(String msg) {
    s = formNameGet(msg, 6); 163
    if (stringEqual(s, "w_ABPO10")) { 154
        w_ABPO10_Set(msg);
    } else if (stringEqual(s, "w_ABPO11")) {
        w_ABPO11_Set(msg);
    } else {
        ...
        ...
        ...
    }
}

```

【図11】

```

String s;
String messageGet(int x, int y, int len);
String messageSet(String msg, int x, int y, int len);
void w_SP_Show(String msg) {
    messageMap(msg);
    w_SP = windowCreate 0;
    {
152
    b_SEND = buttonCreate(w_SP, BUTTON_LOWER_RIGHT,
        "実行");
    buttonAddEvent(b_SEND, BUTTON_CLICK, w_SP_Send);
}

void w_SP_Send 0 {
    String msg;
    SS 153
    msg = messageUnmap 0;
    hostSend(msg);
    windowDelete(w_SP);
    msg = hostReceive 0;
    formSet(msg);
}

```

フロントページの続き

(72)発明者 小泉 忍

神奈川県川崎市麻生区王禅寺1099番地 株
 式会社日立製作所システム開発研究所内